AF/2700

PATENT
ATTY. DOCKET NO. IBM/112
Confirmation No. 6131

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Troy David Armstrong et al.	Art Unit: 2182
Serial No.	:	09/456,211	Examiner: Joshua D. Schneider
Filed	:	December 7, 1999	
For	:	FAIR ELEVATOR SCHEDULING ALGORITHM FOR DIRECT ACCESS STORAGE DEVICE	

Cincinnati, Ohio 45202

June 23, 2003

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF (PATENT APPLICATION-37CFR 191)

1. Transmitted herewith in triplicate is the APPEAL BRIEF in this application with respect to the Notice of Appeal received by the Office on April 24, 2003.

2. **STATUS OF APPLICANT**

This application is on behalf of

XX **other than a small entity**

____ small entity

Verified Statement:

___ attached

___ already filed

3. **FEE FOR FILING APPEAL BRIEF**

Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is:

____	Small entity	$160.00
XX	**Other than a small entity**	**$320.00**

4. **EXTENSION OF TIME**

Applicant petitions for an extension of time under 37 C.F.R. 1.136(a) for the total number of months checked below:

Months		Fee for other than small entity	Fee for small entity
_____	one month	$ 110.00	$ 55.00
_____	two months	..... 400.00	..... 200.00
_____	three months	..... 920.00	..... 460.00
_____	four months	..... 1,440.00	..... 720.00
_____	five months	..... 1,960.00	..... 980.00

Fee: $_____

If an additional extension of time is required, please consider this a petition therefor.

5. **TOTAL FEE DUE**

The total fee due is:

Appeal Brief Fee **$320.00**

Extension Fee _____

6. **FEE PAYMENT**

XX Attached is a check in the sum of **$320.00**.

____ Charge fee to Deposit Account No. 23-3000.

7. **FEE DEFICIENCY**

XX Charge any additional extension fee required or credit any overpayment to Deposit Account No. 23-3000.

WOOD, HERRON & EVANS, L.L.P.

By _____
Scott A. Stinebruner
Reg. No. 38,323

2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
(513) 241-2324

CERTIFICATE OF MAILING 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 23, 2003.

_____
Scott A. Stinebruner Reg. No. 38,323

Attorney Docket No. IBM/112

PATENT

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Troy David Armstrong and Michael Steven Faunce

Appeal No. _____
Application No. 09/456,211

APPEAL BRIEF

PATENT

IBM/112
Confirmation No. 6131

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Troy David Armstrong et al.	Art Unit: 2182
Serial No.:	09/456,211	Examiner: Joshua D. Schneider
Filed:	December 7, 1999	Atty. Docket No.: IBM/112
For:	FAIR ELEVATOR SCHEDULING ALGORITHM FOR DIRECT ACCESS STORAGE DEVICE	

APPEAL BRIEF

Mail Stop APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-29 are currently pending. All claims are as originally filed. Moreover, all claims currently stand rejected, and are now on appeal.

IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to final rejection (Paper No. 7).

V. SUMMARY OF INVENTION

The invention is generally related to an apparatus, program product and method of processing access requests for a Direct Access Storage Device (DASD) utilizing a novel

scheduling algorithm referred to in the Application as a "fair elevator" algorithm to schedule access requests from a plurality of requesters desiring access to a DASD.

In particular, as described at pages 1-2 of the Application, a DASD, e.g., a magnetic, optical, or magneto-optical disk drive or array of such disk drives, is often accessible by multiple users or computer software tasks running in a computer. The access times of a DASD, however, are relatively slow compared to the speed at which the principal processing components of a computer operate, and as such the speed at which data can be transferred to or from a storage device can play a large part in the overall performance of a computer.

Different users and/or different tasks that require access to a DASD (collectively referred to in the Application as "requesters") typically operate concurrently with one another, and often attempt to access the DASD at the same time. To address such concurrent access attempts, various scheduling algorithms have been developed to attempt to ensure that all requests are promptly handled in the most efficient manner possible. However, it has been found that in many instances, it is difficult for a scheduling algorithm to adequately balance the competing interests relating to the performance of the software executing on the computer, and the performance of the DASD.

Specifically, DASD performance is often directly impacted by the "seek time" of a DASD -- that is, the time that it takes for a drive head in the DASD to reach the position on a storage medium of the information with which a particular request is associated. One scheduling algorithm that attempts to minimize seek times is referred to as an "elevator" algorithm, which orders requests based upon the relative positions of the data associated with the requests on the DASD. The ordering of the requests is selected so that a drive head can be moved back and forth between the two end positions of the DASD, with any I/O requests encountered along the way issued to the device. The seek time of the drive head on the DASD is thus minimized, thereby maximizing the performance of the DASD.

As discussed at page 2 of the Application, one potential problem with the elevator algorithm, however, is that if one requester issues a large amount of requests to a relatively narrow range of positions on a DASD, the requester may occupy a large amount of the bandwidth of the DASD. Doing so can effectively restrict access to the DASD for other requesters, thereby stalling the progress of those requesters and adversely impacting system performance.

In other computers, a different scheduling algorithm, often referred to as a "fair" algorithm, attempts to schedule requests in a round robin fashion according to the various identities of the requesters associated with the requests such that each requester in a system is able to use the DASD "fairly", and not unduly stall other requesters attempting to concurrently access the DASD. Multitasking performance is often optimized because multiple tasks are allowed to proceed with reduced delays in accessing a shared DASD. However, given that different requesters will often access data stored at different positions on a DASD, the fair algorithm can increase seek times, and thus degrade the performance of the DASD.

As described at page 4 of the Application, Applicants have addressed the shortcomings of both the elevator algorithm and the fair algorithm through the provision of a "fair elevator" algorithm that schedules access requests to a DASD based upon both the requesters (e.g., the tasks, programs, users, etc. that may issue requests to a shared resource such as a DASD), and the positions on the DASD (e.g., as identified by track, sector, address, cylinder, etc.).

Typically, a fair elevator algorithm consistent with Applicants' invention operates by first sorting pending access requests first on the basis of the requesters associated with the access requests to generate a first ordered set of access requests (Application, page 4, *see also* Fig. 4 and page 11). Then, at least a portion of the access requests in the first ordered set of access requests are sorted based upon the positions on the DASD associated therewith (Application, page 4, *see also* Fig. 5 and pages 11-12). Sorting the first ordered set of access requests creates a second ordered set of access requests, which may be issued in sequence to a DASD to thereby incorporate both requester identity and DASD position into the arbitration of access requests.

In addition, the first and second ordered sets of access requests may be stored in queues, with the first queue (referred to in the Application as a "task queue") storing incoming access requests sorted based upon requester identity, and with the second queue (referred to in the Application as a "position queue") receiving access requests from the first queue and sorted based upon the DASD positions associated therewith (Application, page 4, *see also* Figs. 4-5 and pages 11-12).

By sorting based upon both the requests and the DASD positions associated with different requests, Applicants have addressed a number of the drawbacks associated with conventional elevator and fair algorithms. In particular, Applicants' claimed algorithm looks at

both the requesters and the DASD positions, and as a result, addresses performance concerns both from the perspective of multitasking in the computer and accesses to the storage media of the DASD.

VI. ISSUES

A. Whether claims 1-2, 9-16, and 20-29 were improperly rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,644,786 to *Gallagher et al.* in view of *Applicants' Admitted Prior Art (AAPA).*

B. Whether claims 3-8, and 17-19 were improperly rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,644,786 to *Gallagher et al.* in view of *Applicants' Admitted Prior Art (AAPA)*, and in further view of U.S. Patent No. 5,931,912 to *Wu et al.*

VII. GROUPING OF CLAIMS

Claims 1-29 do not stand or fall together.

VIII. ARGUMENT

Applicants respectfully submit that the Examiner's rejections of claims 1-29 are not supported on the record, and that the rejections should be reversed. Reversal of all rejections, and passage of this case onto allowance, are therefore respectfully requested.

A. Claims 1-2, 9-16, and 20-29 were improperly rejected under 35 U.S.C. § 103(a) as being unpatentable over *Gallagher et al.* in view of *AAPA.*

Applicants respectfully submit that the Examiner's obviousness rejections of claims 1-2, 9-16, and 20-29 based upon U.S. Patent No. 5,644,786 to *Gallagher et al.* in view of *AAPA* are not supported on the record, and that the rejections should be reversed.

A *prima facie* showing of obviousness requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary

skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art. Such a showing also requires objective evidence of the suggestion, teaching or motivation to combine or modify prior art references, as "[c]ombining prior art references without evidence of such a suggestion, teaching or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability -- the essence of hindsight." In re Dembiczak, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

Applicants respectfully submit that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to claims 1-2, 9-16 and 20-29, and as such, the rejections should be reversed. The claims at issue with respect to this rejection are discussed separately below.

Claim 1

Claim 1 generally recites a method of processing access requests for a direct access storage device (DASD) such as a disk drive, where each access request is associated with a requester and a position on the DASD. The method includes sorting at least a subset of a plurality of access requests directed to the DASD based upon the requesters associated therewith to generate a first ordered set of access requests, sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests, and issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

As such, claim 1 incorporates, in part, the concept of sorting access requests directed to a DASD based upon both the requesters and the DASD positions associated with such access requests.

The Examiner rejects claim 1 as being obvious over *Gallagher et al.* in view of *AAPA*, essentially arguing that both sorting requests based on requesters and sorting requests based on positions are individually known, and that it would have been obvious to combine these sorting techniques into a single algorithm. Applicants respectfully disagree with the Examiner's conclusions, however, as the Examiner has failed to establish any recognized motivation in the art for combining these techniques in the manner recited in claim 1. Absent any such motivation, the rejection is based upon hindsight, and must be reversed.

As described above, Applicants' claimed invention focuses on a novel scheduling algorithm that has been referred to in the Application as a "fair elevator" algorithm. In doing so, both the requesters (e.g., the tasks, programs, users, etc. that may issue requests to a shared resource such as a DASD), as well as the positions on the DASD (e.g., as identified by track, sector, address, cylinder, etc.), that are associated with pending access requests, are used to sort those access requests for submission to the DASD.

Admittedly, sorting access requests to a DASD based upon requester, and sorting access requests to a DASD based upon DASD position, are individually known in the art. The former type of sorting, referred to in the art as a "fair" algorithm, focuses on maximizing multitasking performance in a computer by sorting requests based upon their associated requesters, *i.e.,* to ensure that multiple requesters are able to proceed at a reasonable rate when those requesters are accessing the same shared resource. The latter type of sorting, referred to in the art as an "elevator" algorithm, focuses on the performance of the DASD itself, by minimizing seek times through the ordered arrangement of requests based upon DASD position.

However, Applicants have found that, while each algorithm individually can benefit system performance, each algorithm is subject to certain drawbacks that mitigate the performance benefits obtainable thereby. The "fair" algorithm, for example, can suffer from inefficient DASD performance due to the need for the DASD to jump around to access data stored at different positions when different requesters are accessing data at different regions of the DASD. Likewise, the "elevator" algorithm can cause certain requesters to become stalled if one particular requester issues many requests to the same general region on a DASD.

Applicants' claimed algorithm therefore addresses these shortcomings by looking at both the requesters and the DASD positions associated with individual access requests. By doing so, both multitasking performance from the perspective of the data processing system, and DASD performance from the perspective of a shared resource (i.e., a DASD) can each be improved without the attendant negative impact on the other.

In rejecting claim 1, the Examiner relies on *Gallagher et al.* for disclosing an elevator algorithm, and on *AAPA* for disclosing a fair algorithm. Where the Examiner's rejection falls short of establishing a *prima facie* case of obviousness of claim 1, however, is in that the rejection fails to establish through objective evidence any suggestion or motivation in the prior

art to modify the elevator algorithm disclosed in *Gallagher et al.* to address fairness concerns by additionally considering the identity of the requesters associated with the requests being sorted by the *Gallagher et al.* algorithm. Absent such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection therefore should be reversed.

Gallagher et al. itself certainly provides no objective evidence of motivation. Specifically, *Gallagher et al.*, as noted by the Examiner, discusses only a conventional elevator algorithm, *i.e.*, an algorithm that sorts requests based upon position. Moreover, there is no disclosure or suggestion anywhere in the reference of the desirability of addressing "fairness" with respect to multiple requesters of a shared resource.

In support of the rejection, the Examiner notes in ¶3 of the Final Office Action that *Gallagher et al.* teaches the use of two queues to sort requests. However, it is also important to note that both queues are used to sort based upon position-related criteria: the first queue sorts by sector number, while the second queue sorts by cylinder number (*Gallagher et al.*, col. 4, lines 1-44). Thus, the fact that the reference utilizes two queues is of little relevance, as "fairness" considerations (*i.e.*, sorting by requester) are still not addressed in either queue.

In addition, the Examiner notes in ¶31 of the Final Office Action that *Gallagher et al.* discloses the stacking of requests based upon priority. Col. 4, lines 44-51 of *Gallagher et al.*, in particular, discloses that "packets" of requests that are forwarded to the execution queue may include requests having the same priority, and that higher priority packets may be placed in the execution queue ahead of lower priority packets. However, the priority of a request has nothing to do with the identity of the requester associated with the request, so the fact that priority may be considered when moving requests between queues is also insufficient to establish the required motivation to incorporate fairness into a conventional elevator algorithm.

Even if the priority of a request was somehow analogized to the identity of a requester for a request, however, the rejection would still be deficient with regard to claim 1, as the claim sorts requests first by requester, and second by position. *Gallagher et al.* discloses that requests are sorted by a position-based criterion (sector number) in the first queue (col. 4, lines 1-7), and then moved from the first queue to the second queue in packets grouped by priority (col. 4, lines 44-51). Thus, even if the second queue was considered to be a non-position-based queue, this

configuration would be precisely opposite of that recited in claim 1, where a non-position-based criterion (requester) is used to sort requests prior to sorting by a position-based criterion. Given that *Gallagher et al.* would disclose the opposite scenario to that recited in claim 1 under this interpretation, the Examiner cannot argue that *Gallagher et al.* suggests sorting by priority or any other non-position criterion prior to sorting by position.

Furthermore, the only way in which priority could be analogized to the identity of a requester would be if all requests issued by various requesters had the same priority (*i.e.*, requester 1 requests are priority "1", requester 2 requests are priority "2", etc.). In such a scenario, however, fairness would most certainly not be addressed by the *Gallagher et al.* algorithm, since priorities are used to always issue higher priority requests before lower priority requests (*Gallagher et al.*, col. 4, lines 44-51). The *Gallagher et al.* algorithm would therefore have the same, if not a greater, risk of unfairly favoring one requester over another than even a pure elevator algorithm, since all of the requests for one requester would always be issued before issuing any requests for lower priority requesters.

As such, attempting to analogize priority with requester identity, as apparently done so by the Examiner, would result in an interpretation of *Gallagher et al.* that goes as far as teaching away from the claimed invention. The fact that *Gallagher et al.* recognizes the desirability of grouping by priority is therefore of little importance to the patentability of claim 1.

Applicants therefore respectfully submit that *Gallagher et al.* cannot be relied upon to provide the required objective evidence of motivation.

Likewise, the *AAPA* fails to provided the necessary evidence of motivation. Applicants have made no admission in the Application as to the conventionality of combining the "fair" and "elevator" algorithms together. Indeed, Applicants' disclosure as a whole argues that it is the combination of these two techniques into the claimed "fair elevator" algorithm that substantially contributes to distinguishing the invention from the prior art. Any attempt to read such motivation into the Application would expressly run counter to Applicants' disclosure as a whole, and therefore would not be proper.

As an additional matter, Applicants respectfully submit that reliance on Applicants' disclosure to provide the required evidence of the suggestion, teaching or motivation to modify *Gallagher et al.* would necessarily constitute the use of Applicants' disclosure as a "blueprint" for

modifying the prior art - the essence of hindsight-based analysis. Applicants respectfully submit that the Examiner must be able to cite some evidence of suggestion or motivation that is independent of Applicants' disclosure to support an obviousness rejection. The Examiner has not done so in this case.

Given that neither *Gallagher et al.*, nor the *AAPA*, can be relied upon to provide the required evidence of motivation, the Examiner must be able to point to some other teaching in the prior art that suggests the desirability of combining "fairness" techniques (relying on the requesters associated with access requests) with "elevator" techniques (relying on the DASD positions associated with access requests) in a common algorithm. In this case, the Examiner has not pointed to, nor could the Examiner point to, any other teaching in the prior art that suggests this combination.

Absent a citation of clear, objective evidence as to the desirability of combining fairness and elevator techniques in a common algorithm, the Examiner's rejection amounts to nothing more than hindsight-based reasoning. Applicants therefore respectfully submit that the Examiner has failed to raise a *prima facie* case of obviousness as to claim 1. Reversal of the Examiner's rejection of claim 1, as well as allowance of the claim, are therefore respectfully requested.

Claim 2

Claim 2 is not separately argued.

Claim 9

Claim 9 recites a method of processing access requests for DASD, where each access request is associated with a requester and a position on the DASD. The claimed method includes sorting a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith, and issuing the sorted access requests to the DASD.

As with claim 1, claim 9 is non-obvious over *Gallagher et al.* and the *AAPA*. As discussed above in connection with claim 1, neither *Gallagher et al.*, nor the *AAPA*, disclose or suggest a sorting algorithm that relies on both the requesters and positions associated with each access request. Moreover, the Examiner has provided no objective evidence that would motivate

one of ordinary skill in the art to modify the elevator algorithm disclosed in *Gallagher et al.* to also sort access requests based upon the requesters associated with such access requests.

Absent any such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection of the claim should be reversed. Reversal of the Examiner's rejection of claim 9, as well as allowance of the claim, are therefore respectfully requested.

Claim 10

Claim 10 depends from claim 9, and additionally recites, *inter alia*, sorting access requests based upon the requesters associated therewith to generate a first ordered set of access requests, and sorting access requests in the first ordered set based upon the positions associated therewith.

As discussed above in connection with claim 1, even under an excessively-broad reading of *Gallagher et al.*, where ordering by priority might be analogized to sorting based upon requesters, *Gallagher et al.* could only be read to suggest at the most sorting based upon position-based criterion, followed by sorting based upon a non-position-based criterion – precisely the opposite scenario recited in claim 10. Applicants respectfully submit that *Gallagher et al.* could not be read to suggest an opposite scenario from that disclosed in the reference, and as such, could not be read to supply the motivation necessary to sustain an obviousness rejection. Reversal of the Examiner's rejection of claim 10, as well as allowance of the claim, are therefore respectfully requested.

Claim 11

Claim 11 is not separately argued.

Claim 12

Claim 12 recites a method of processing access requests for a DASD, which includes *inter alia* receiving incoming access requests into a first queue that is sorted based upon the requester associated with each access request stored in the first queue, and moving at least one

access request from the first queue to a second queue that is sorted based upon the position associated with each access request stored in the second queue.

As with claim 1, claim 12 is non-obvious over *Gallagher et al.* and the *AAPA,* as neither *Gallagher et al.*, nor the *AAPA*, disclose or suggest a sorting algorithm that sorts requests based upon requester (or any other non-position-based criterion), followed by sorting requests based upon position. Moreover, the Examiner has provided no objective evidence that would motivate one of ordinary skill in the art to modify the elevator algorithm disclosed in *Gallagher et al.* to also sort access requests based upon the requesters associated with such access requests, prior to sorting based upon position.

Absent any such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection of the claim should be reversed. Reversal of the Examiner's rejection of claim 12, as well as allowance of the claim, are therefore respectfully requested.

Claim 13

Claim 13 recites an apparatus which includes *inter alia* a program configured to sort a plurality of access requests directed to a DASD based upon both the requesters and the positions associated therewith.

As with claim 1, claim 13 is non-obvious over *Gallagher et al.* and the *AAPA,* as neither *Gallagher et al.*, nor the *AAPA*, disclose or suggest a sorting algorithm that relies on both the requesters and positions associated with each access request. Moreover, the Examiner has provided no objective evidence that would motivate one of ordinary skill in the art to modify the elevator algorithm disclosed in *Gallagher et al.* to also sort access requests based upon the requesters associated with such access requests.

Absent any such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection of the claim should be reversed. Reversal of the Examiner's rejection of claim 13, as well as allowance of the claim, are therefore respectfully requested.

Claim 14

Claim 14 is not separately argued.

Claim 15

Claim 15 depends from claim 13, and additionally recites, *inter alia*, sorting access requests based upon the requesters associated therewith to generate a first ordered set of access requests, and sorting access requests in the first ordered set based upon the positions associated therewith.

As discussed above in connection with claim 1, even under an excessively-broad reading of *Gallagher et al.*, where ordering by priority might be analogized to sorting based upon requesters, *Gallagher et al.* could only be read to suggest at the most sorting based upon position-based criterion, followed by sorting based upon a non-position-based criterion – precisely the opposite scenario recited in claim 15. Applicants respectfully submit that *Gallagher et al.* could not be read to suggest an opposite scenario from that disclosed in the reference, and as such, could not be read to supply the motivation necessary to sustain an obviousness rejection. Reversal of the Examiner's rejection of claim 15, as well as allowance of the claim, are therefore respectfully requested.

Claims 16 and 20-22

Claims 16 and 20-22 are not separately argued.

Claim 23

Claim 23 recites, *inter alia*, an apparatus that includes a first queue configured to receive incoming access requests that are sorted based upon the requesters associated therewith, and a second queue configured to issue outgoing access requests to the DASD that are sorted based upon the positions associated therewith. The claimed apparatus also includes control logic that moves access requests from the first queue to the second queue.

As with claim 1, claim 23 is non-obvious over *Gallagher et al.* and the *AAPA,* as neither *Gallagher et al.*, nor the *AAPA*, disclose or suggest a sorting algorithm that sorts requests based upon requester (or any other non-position-based criterion), followed by sorting requests based

upon position. Moreover, the Examiner has provided no objective evidence that would motivate one of ordinary skill in the art to modify the elevator algorithm disclosed in *Gallagher et al.* to also sort access requests based upon the requesters associated with such access requests, prior to sorting based upon position.

Absent any such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection of the claim should be reversed. Reversal of the Examiner's rejection of claim 23, as well as allowance of the claim, are therefore respectfully requested.

Claims 24-27

Claims 24-27 are not separately argued.

Claim 28

Claim 28 recites a program product, which includes *inter alia* a program configured to sort a plurality of access requests directed to a DASD based upon both the requesters and the positions associated therewith.

As with claim 1, claim 28 is non-obvious over *Gallagher et al.* and the *AAPA,* as neither *Gallagher et al.*, nor the *AAPA*, disclose or suggest a sorting algorithm that relies on both the requesters and positions associated with each access request. Moreover, the Examiner has provided no objective evidence that would motivate one of ordinary skill in the art to modify the elevator algorithm disclosed in *Gallagher et al.* to also sort access requests based upon the requesters associated with such access requests.

Absent any such objective evidence, the Examiner's rejection amounts to nothing more than hindsight-based reasoning, and the rejection of the claim should be reversed. Reversal of the Examiner's rejection of claim 28, as well as allowance of the claim, are therefore respectfully requested.

Claim 29

Claim 29 is not separately argued.

B. Claims 3-8, and 17-19 were improperly rejected under 35 U.S.C. § 103(a) as being unpatentable over *Gallagher et al.* in view of *AAPA*, and in further view of *Wu et al.*

Applicants respectfully submit that the Examiner's obviousness rejections of claims 3-8 and 17-19 based upon *Gallagher et al.* in view of *AAPA*, and further in view of U.S. Patent No. 5,931,912 to *Wu et al.* are also not supported on the record, and that the rejections should be reversed. The claims at issue with respect to this rejection are discussed separately below.

Claim 3

Claim 3 depends from claim 2, and additionally recites that sorting the access requests received into the first queue includes, for each inbound access request, determining whether another access request in the first queue is associated with the same requester as the inbound access request; if so, storing the inbound access request after a last access request associated with the same requester as the inbound access request; and, if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

In rejecting claim 3, the Examiner is forced to rely on *Wu et al.* for allegedly teaching sorting by requestor to include an attempt to match a new requestor with a requestor currently waiting to be processed. Presumably, the Examiner does not consider the additional features in claim 3 to be disclosed by either *Gallagher et al.* or the *AAPA*.

Wu et al., on the other hand, is directed to a method of tracking user accesses to hypertext objects available on a web server. The Examiner in particular relies on a logging mechanism that stores accesses to hypertext objects by different users, and which attempts to determine if an access is related to any other accesses by the same user, and if so, group the access in the log with the other accesses. *See, e.g.,* col. 6, line 52 to col. 7, line 13.

While the Examiner admits, in ¶21 of the Final Office Action, that *Wu et al.* is not used for providing fair and efficient access to a drive, the Examiner attempts to formulate at least a tenuous connection between *Wu et al.* and the subject matter of claim 3 by arguing that the *Wu et al.* method is used in a system to allow accesses to a drive to be analyzed.

The Examiner's attempt notwithstanding, *Wu et al.* appears to have no relevance whatsoever to the subject matter of claim 3. In this instance, *Wu et al.* is so irrelevant as to both the technological field of endeavor, and the specific problem being addressed, that Applicants

submit that one of ordinary skill in the art would not look to *Wu et al.* to solve the problem addressed by claim 3.

Other than being used in a computer, the *Wu et al.* method has no relevance to the submission of access requests to a DASD or other form of storage device. While *Wu et al.* speaks of requests, it is readily apparent from the reference that the requests refer to network requests, *e.g.*, to access Internet content from a web server. The Board will appreciate that computers commonly communicate using data or signals that may be characterized as "requests", so much so that likely a multitude of different types of communications that are entirely unrelated to one another may all be characterized as "requests." Simply the fact that *Wu et al.* discusses requests does not make those requests relevant to the particular type of access requests referred to in claim 3.

Likewise, the fact that Internet content is stored on DASD's does not make network requests to access that content analogous to an access request within the context of Applicants' invention. Claim 3 refers to requests that are scheduled according to a scheduling algorithm to access a DASD, a concept that is entirely foreign to *Wu et al.*

Furthermore, the purpose of organizing requests in *Wu et al.* is to order log entries to assist in analyzing user access patterns to hypertext content, and not to order requests prior to delivery to a storage medium. As such, *Wu et al.* appears to have no relevance to claim 3. Applicants therefore respectfully submit that claim 3 is non-obvious over *Wu et al.*, as well as *Gallagher et al.* and the *AAPA*. Reversal of the Examiner's rejection of claim 3, as well as allowance of the claim, are therefore respectfully requested.

Claims 4-8

Claims 4-8 are not separately argued.

Claim 17

Claim 17 depends from claim 16, and additionally recites *inter alia* that sorting the access requests received into the first queue includes, for each inbound access request, determining whether another access request in the first queue is associated with the same requester as the inbound access request; if so, storing the inbound access request after a last access request

associated with the same requester as the inbound access request; and, if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

As with claim 3, *Wu et al.* would not be looked to by one of ordinary skill in the art to solve the problem addressed by claim 17. *Wu et al.* is directed to a different technological field, and addresses a different solution to a different problem. Applicants therefore respectfully submit that claim 17 is non-obvious over *Wu et al.*, as well as *Gallagher et al.* and the *AAPA*. Reversal of the Examiner's rejection of claim 17, as well as allowance of the claim, are therefore respectfully requested.

Claims 18-19

Claims 18-19 are not separately argued.

IX. CONCLUSION

In conclusion, Applicants respectfully request that the Board reverse the Examiner's rejections of claims 1-29, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324. Moreover, if any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

WOOD, HERRON & EVANS, L.L.P.

Date: 23 JUNE 2003

By: _____
Scott A. Stinebruner
Reg. No. 38,323

2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
(513) 241-2324

APPENDIX A: CLAIMS ON APPEAL (S/N 09/456,211)

1. (Unchanged) A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) sorting at least a subset of a plurality of access requests directed to the DASD based upon the requesters associated therewith to generate a first ordered set of access requests;

(b) sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests; and

(c) issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

2. (Unchanged) The method of claim 1, further comprising moving access requests between a first queue and a second queue, wherein sorting the access requests to generate the first ordered set of access requests includes receiving inbound access requests into the first queue, and sorting the access requests received into the first queue based upon the requesters associated therewith, wherein sorting the access requests in the first ordered set includes sorting access requests moved into the second queue based upon the positions associated therewith, and wherein issuing the access requests in the second ordered set includes removing issued access requests from the second queue.

3. (Unchanged) The method of claim 2, wherein sorting the access requests received into the first queue includes, for each inbound access request:

(a) determining whether another access request in the first queue is associated with the same requester as the inbound access request;

(b) if so, storing the inbound access request after a last access request associated with the same requester as the inbound access request; and

(c) if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

4. (Unchanged) The method of claim 3, wherein moving access requests between the first queue and the second queue is performed in response to determining that the second queue is empty.

5. (Unchanged) The method of claim 4, wherein moving access requests between the first queue and the second queue includes, for each requester associated with an access request stored in the first queue, moving at least one access request associated with such requester into the second queue.

6. (Unchanged) The method of claim 2, wherein moving access requests between the first queue and the second queue further includes moving a batch of access requests between the first queue and second queue, and wherein sorting the access requests moved into the second queue includes reversing a sort order for each successive batch of access requests.

7. (Unchanged) The method of claim 1, wherein sorting the access requests in the first ordered set includes alternately sorting access requests in ascending and descending order.

8. (Unchanged) The method of claim 1, wherein each access request is associated with one of a plurality of requesters in a computer coupled to the DASD, with each requester being a computer task executing on the computer.

9. (Unchanged) A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) sorting a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith; and

(b) issuing the sorted access requests to the DASD.

10. (Unchanged) The method of claim 9, wherein sorting the plurality of access requests includes:

(a) sorting at least a subset of a plurality of access requests based upon the requesters associated therewith to generate a first ordered set of access requests; and

(b) sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests, wherein issuing the sorted access requests includes issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

11. (Unchanged) The method of claim 9, wherein sorting the plurality of access requests includes:

(a) sorting at least a subset of a plurality of access requests using one of a fair scheduling algorithm and an elevator scheduling algorithm to generate a first ordered set of access requests; and

(b) sorting at least a subset of the access requests in the first ordered set of access requests using the other of the fair scheduling algorithm and the elevator scheduling algorithm.

12. (Unchanged) A method of processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the method comprising:

(a) receiving incoming access requests into a first queue that is sorted based upon the requester associated with each access request stored in the first queue;

(b) moving at least one access request from the first queue to a second queue that is sorted based upon the position associated with each access request stored in the second queue; and

(c) sequentially issuing access requests from the second queue to the DASD.

13. (Unchanged) An apparatus for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the apparatus comprising:

(a) a memory; and

(b) a program, resident in the memory, the program configured to sort a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith, and issue the sorted access requests to the DASD.

14. (Unchanged) The apparatus of claim 13, wherein the program is configured to sort the plurality of access requests by sorting at least a subset of a plurality of access requests using one of a fair scheduling algorithm and an elevator scheduling algorithm to generate a first ordered set of access requests; and sorting at least a subset of the access requests in the first ordered set of access requests using the other of the fair scheduling algorithm and the elevator scheduling algorithm.

15. (Unchanged) The apparatus of claim 13, wherein the program is configured to sort the plurality of access requests by sorting at least a subset of a plurality of access requests based upon the requesters associated therewith to generate a first ordered set of access requests, and sorting at least a subset of the access requests in the first ordered set of access requests based upon the positions associated therewith to generate a second ordered set of access requests, and wherein the program is configured to issue the sorted access requests by issuing each of the access requests in the second ordered set of access requests in sequence to the DASD.

16. (Unchanged) The apparatus of claim 15, wherein the program is further configured to:

(a) move access requests between a first queue and a second queue;

(b) sort the access requests to generate the first ordered set of access requests by receiving inbound access requests into the first queue and sorting the access requests received into the first queue based upon the requesters associated therewith;

(c) sort the access requests in the first ordered set by sorting access requests moved into the second queue based upon the positions associated therewith; and

(d) issue the access requests in the second ordered set by removing issued access requests from the second queue.

17. (Unchanged) The apparatus of claim 16, wherein the program is configured to sort the access requests received into the first queue by, for each inbound access request:

(a) determining whether another access request in the first queue is associated with the same requester as the inbound access request;

(b) if so, storing the inbound access request after a last access request associated with the same requester as the inbound access request; and

(c) if not, arranging the inbound access request within the first queue based upon a requester identifier associated therewith.

18. (Unchanged) The apparatus of claim 17, wherein the program is configured to move access requests between the first queue and the second queue in response to determining that the second queue is empty.

19. (Unchanged) The apparatus of claim 18, wherein the program is configured to move access requests between the first queue and the second queue by, for each requester associated with an access request stored in the first queue, moving at least one access request associated with such requester into the second queue.

20. (Unchanged) The apparatus of claim 16, wherein the program is configured to move access requests between the first queue and the second queue by moving a batch of access requests between the first queue and second queue, and wherein the program is configured to sort the access requests moved into the second queue by reversing a sort order for each successive batch of access requests.

21. (Unchanged) The apparatus of claim 15, wherein the program is configured to sort the access requests in the first ordered set by alternately sorting access requests in ascending and descending order.

22. (Unchanged) The apparatus of claim 15, further comprising a plurality of requester computer tasks resident in the memory, wherein each access request is associated with one of the plurality of requester computer tasks.

23. (Unchanged) An apparatus for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the apparatus comprising:

(a) a first queue configured to receive incoming access requests, the access requests in the first queue sorted based upon the requesters associated therewith;

(b) a second queue configured to issue outgoing access requests to the DASD, the access requests in the second queue sorted based upon the positions associated therewith; and

(c) control logic coupled to the first and second queues and configured to selectively move access requests from the first queue to the second queue.

24. (Unchanged) The apparatus of claim 23, further comprising a memory; and a processor coupled to the memory, wherein the control logic comprises request processing program code resident in the memory.

25. (Unchanged) The apparatus of claim 24, further comprising an operating system resident in the memory, wherein the request processing program code and the first and second queues are resident in the operating system.

26. (Unchanged) The apparatus of claim 25, wherein the operating system includes a DASD hardware driver that interfaces the apparatus with the DASD, and wherein the request processing program code and the first and second queues are resident in the DASD hardware driver.

27. (Unchanged) The apparatus of claim 24, wherein the first and second queues are resident in the memory.

28. (Unchanged) A program product, comprising:

(a) a program for use in processing access requests for a direct access storage device (DASD), each access request associated with a requester and a position on the DASD, the program configured to sort a plurality of access requests directed to the DASD based upon both the requesters and the positions associated therewith, and issue the sorted access requests to the DASD; and

(b) a signal bearing medium bearing the program.

29. (Unchanged) The program product of claim 28, wherein the signal bearing medium includes at least one of a recordable medium and a transmission type medium.